

VI. WHAT IS CLAIMED

Sub A 7

1. A computer-implemented method for inlining code of a computer program, comprising:

identifying a subprogram of the computer program; and

5 selectively inlining computer code of certain execution paths of the subprogram.

2. The method of claim 1 comprising identifying the subprogram based on execution characteristics of the subprogram.

3. The method of claim 1 wherein said step of selectively inlining is based on execution characteristics of the execution paths.

4. The method of claim 3 wherein the execution characteristics are based on the execution time for the paths.

5. The method of claim 4 wherein the execution characteristics are based on the frequency of execution of the paths.

6. The method of claim 1 wherein the step of selectively inlining is based on an inline indication associated with an execution path.

7. The method of claim 6 wherein the inline indication is an inline directive.

8. The method of claim 7 wherein the inline directive is included as part of a program comment statement.

9. The method of claim 1 further comprising determining whether to selectively inline the execution paths using information profiles associated with the execution path.

Sub A 7 10. A computer program compiler for inlining computer program code, comprising:

a subprogram identification module that identifies subprograms of the computer code; and

a path identification module that selectively inlines computer code of certain execution paths of the subprogram.

11. The apparatus of claim 10 wherein the path identification module selectively inlines based on an inline indication associated with an execution path.

12. The apparatus of claim 11 wherein the inline indication is an inline directive.

13. The method of claim 12 wherein the inline directive is included as part of a program comment statement.

Sub A 7 14. A computer-implemented method of determining whether to replace subprogram code of a computer program, comprising the steps of:

identifying a subprogram that has a first and a second execution characteristic;

replacing a portion of the subprogram that exhibits the first execution characteristic with program instructions that explicitly define the operations of the first execution characteristic; and

Sub A' 7

leaving intact a second portion of the subprogram that exhibits the second execution characteristic.

15. The method of claim 14 wherein said second execution characteristic is an atypical characteristic.

5 16. The method of claim 15 wherein said atypical characteristic is an execution time duration.

17. The method of claim 16 wherein said execution time duration exceeds a predetermined threshold.

10 18. The method of claim 17 wherein said first execution characteristic is a typical execution characteristic.

19. The method of claim 17 wherein said first and second execution characteristics are execution time durations.

20. The method of claim 19 wherein the first and second execution characteristics are based on arguments operated on by the subprogram.

15 Sub A' 7 21. The method of claim 20 wherein the first and second execution characteristics are invoked based on conditional execution computer statements associated with the characteristics.

20 22. The method of claim 11 wherein the first and second execution characteristics are invoked based on conditional execution computer statements associated with the characteristics.

23. A computer readable medium for inlining computer program code, which when executed by a computer, performs the steps of:

Sub A7

identifying a subprogram that has a plurality of execution characteristics;  
inlining only a selected portion of the subprogram that corresponds to one of  
the execution characteristics.

5

24. The medium of claim 23 wherein the selected portion is defined by a  
selected path of a plurality of execution paths that may be executed by the  
subprogram.

A

25. The medium of claim 24 further comprising determining the selected  
path by identifying a directive associated with the path.

Sub A7

10

26. The medium of claim 25 wherein the directive is included in the  
selected path.

27. The medium of claim 23 wherein the execution characteristics are  
identified by evaluating a conditional execution statement associated with a  
subprogram call.

15

28. A computer-implemented method of replacing subprogram code in a  
computer system, comprising the steps of:

identifying a subprogram that operates in a first manner when operands  
passed to the subprogram fall within a first range of values and that operates in a  
second manner when operands passed to the subprogram fall within a second  
range of values; and

20

replacing subprogram statements that cause the subprogram to operate in  
the first manner with expanded code.

29. A computer-implemented method for compiling a program containing at least one reference to a subprogram from source code to object code, wherein the subprogram has alternative paths of execution, the method comprising:

5 a) determining the execution path of the subprogram to be executed when the reference to the subprogram is encountered during compilation of the program; and

10 b) replacing the reference to the subprogram in the program with at least a portion of the subprogram corresponding to the execution path of the subprogram to be executed when the reference to the subprogram is encountered during compilation of the program.

30. The method of claim 29 wherein determining the execution path of the subprogram to be executed when the reference to the subprogram is encountered during compilation of the program, comprises:

15 analyzing a set of input parameters that may be provided to the subprogram during execution.

31. The method of claim 29, further comprising:

determining whether the subprogram contains a command signaling a preference for optimizing compilation of the program by performing steps a) and b).

32. The method of claim 29, further comprising:

20 converting the program from source code to object code.

33. A computer-implemented method for compiling a program containing at least one reference to a subprogram from source code to object code, wherein the subprogram has alternative paths of execution, the method comprising:

determining the execution path of the subprogram to be executed when the reference to the subprogram is encountered during compilation of the program; and

replacing the reference to the subprogram with only a portion of the subprogram corresponding to the execution path of the subprogram to be executed when the reference to the subprogram is encountered during compilation of the program.

34. The method of claim 33 wherein determining the execution path of the subprogram to be executed when the reference to the subprogram is encountered during execution of the program, comprises:

analyzing a set of input parameters that may be provided to the subprogram during execution.

35. An apparatus having a processor and a memory containing programs for inlining code of a computer program which when executed using the processor perform steps comprising:

identifying a subprogram of the computer program; and

selectively inlining computer code of certain execution paths of the subprogram.

36. The apparatus of claim 35 wherein said subprogram is identified based on execution characteristics of the subprogram.

Sub A7

37. The apparatus of claim 35 wherein said computer code is selectively inlined is based on execution characteristics of the execution paths.

38. The apparatus of claim 37 wherein the execution characteristics are based on the execution time for the paths.

39. The apparatus of claim 37 wherein the execution characteristics are based on the frequency of execution of the paths.

40. The apparatus of claim 35 wherein the computer code is selectively inlined based on an inline indication associated with an execution path.

41. The apparatus of claim 40 wherein the inline indication is an inline directive.

42. The apparatus of claim 41 wherein the inline directive is included as part of a program comment statement.

43. The apparatus of claim 35 further comprising using information profiles associated with the execution path to determine whether to selectively inline the execution paths.

44. An apparatus having a processor and a memory containing programs for determining whether to replace subprogram code of a computer program which when executed using the processor perform steps comprising:

identifying a subprogram that has a first and a second execution characteristic;

Sub A<sup>1</sup> 7

replacing a portion of the subprogram that exhibits the first execution characteristic with program instructions that explicitly define the operations of the first execution characteristic; and

leaving intact a second portion of the subprogram that exhibits the second execution characteristic.

45. The apparatus of claim 44 wherein said second execution characteristic is an atypical characteristic.

46. The apparatus of claim 45 wherein said atypical characteristic is an execution time duration.

47. The apparatus of claim 46 wherein said execution time duration exceeds a predetermined threshold.

48. The apparatus of claim 47 wherein said first execution characteristic is a typical execution characteristic.

49. The apparatus of claim 47 wherein said first and second execution characteristics are execution time durations.

50. The apparatus of claim 49 wherein the first and second execution characteristics are based on arguments operated on by the subprogram.

Sub K<sup>1</sup> 7

51. The apparatus of claim 50 wherein the first and second execution characteristics are invoked based on conditional execution computer statements associated with the characteristics.



Sub A<sup>1</sup> 7

52.

The apparatus of claim 51 wherein the first and second execution characteristics are invoked based on conditional execution computer statements associated with the characteristics.

53. An apparatus having a processor and a memory containing programs

5

for replacing subprogram code in a computer system which when executed using the processor perform steps comprising, comprising the steps of:

identifying a subprogram that operates in a first manner when operands passed to the subprogram fall within a first range of values and that operates in a second manner when operands passed to the subprogram fall within a second range of values; and

10

replacing subprogram statements that cause the subprogram to operate in the first manner with expanded code.